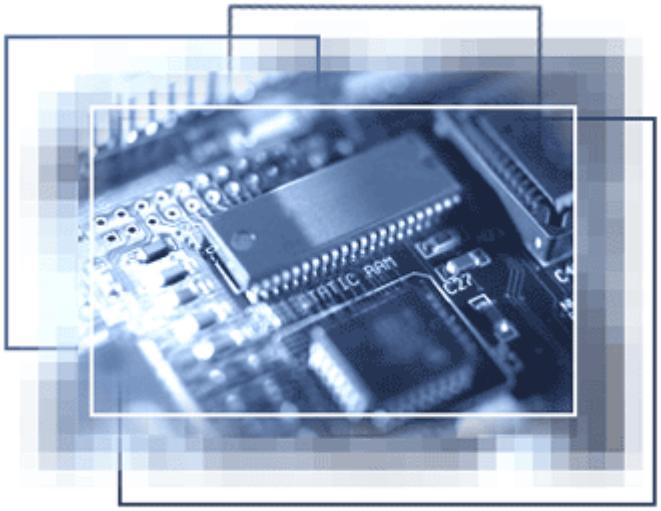
DISEÑO Y CONSTRUCCIÓN DE UN AUTÓMATA PROGRAMABLE PARA APLICACIONES INDUSTRIALES



Autores: Iván Antolínez García

Javi Castañeda Aguilar

Curso: 2º de Bachillerato, modalidad tecnología

Centro: Francesc Xavier Lluch i Rafecas

Localidad: Vilanova i la Geltrú, Barcelona

Tutor del trabajo: Miquel Pascual Cabo

Fecha de entrega: 29 de marzo de 2008

Sumario

1. Introducción al microcontrolador PIC 16F84A.

Podríamos definir la palabra microcontrolador como un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

Los microcontroladores PIC son fabricados por la empresa Microchip Technology Inc. y son de los circuitos integrados más utilizados en cualquiera de los ámbitos tecnológicos.

El microcontrolador PIC16F84A es uno de los PICs más utilizados, debido a sus características que permiten su adaptación en una infinidad de circuitos y aplicaciones.

Dispone de dieciocho patas o *pins*, de los cuales trece se pueden configurar como entradas o salidas de datos.

Una memoria FLASH de 1000 x 14 bits se encarga de almacenar el programa que tendrá que ejecutar y otras dos memorias, EEPROM de 64 x 8 *bits* y RAM de 67 x 8 *bits* se encargan de guardar los datos y variables.

Para determinar la velocidad a la que debe trabajar, el PIC16F84A requiere de un circuito oscilador externo, que puede enviar una frecuencia máxima de 20MHz que será dividida entre cuatro por la frecuencia interna del reloj del PIC. De esta manera, con una frecuencia externa de 4MHz, el microcontrolador ejecuta mil instrucciones por segundo.

Una de las características más importantes de este integrado, es la capacidad que tiene de atender interrupciones. Esto quiere decir que es capaz de, mientras está reproduciendo un programa, ejecutar una subrutina y volver otra vez al programa que estaba reproduciendo. Un claro ejemplo sería el programa de una máquina de refrescos. En el *display* aparece de un lado al otro un mensaje de bienvenida. Pulsamos el botón de la bebida que queremos, aparece el precio de ésta, y al dejar de pulsar el botón, continúa desplazándose el mensaje de bienvenida.

Conocidas las características más importantes del microcontrolador PIC16F84A, podemos profundizar de una manera más amplia en los campos que más nos interesan.

1.1. Terminales.

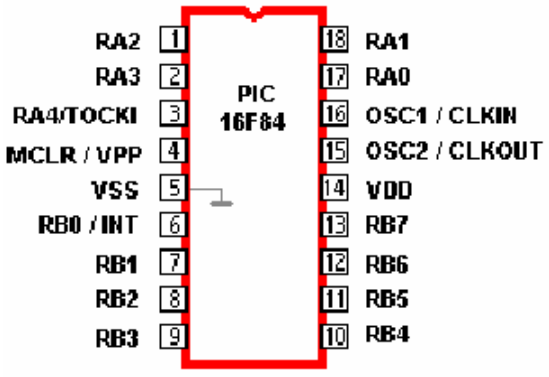


Figura 1: Terminales del PIC 16F84A.

Patas 1, 2, 3, 17 y 18 (RA0-RA4/TOCKI): Corresponden al PUERTO A. Son 5 líneas bidireccionales definidas por programación. El *pin* RA4/TOCKI, como entrada, se puede programar como entrada normal o como entrada del contador/temporizador TMR0.

Pata 4 (MCLR/ Vpp): Esta pata tiene dos aplicaciones. Por un lado es la entrada de reset. Cuando no le llega tensión el PIC hace un reset, y cuando su valor de tensión es el mismo que el de VDD, el PIC funciona correctamente. Por otra parte es la pata que habilita la tensión mientras se programa el PIC.

Patas 5 y 14 (VSS y VDD): Son respectivamente las patas de *masa* y alimentación del PIC. La tensión de alimentación del PIC está comprendida entre 2 y 5,5 voltios.

Patas 6, 7, 8, 9, 10, 11, 12 y 13 (RB0-RB7): Estas patas corresponden al PUERTO B. Son 8 líneas bidireccionales definidas por programación. RB0 puede programarse como entrada de interrupciones externas. Los *pines* de RB4 a RB7 se pueden programar para responder a interrupciones por cambio de estado. Los *pines* RB6 y RB7, a la hora de programar el PIC, se corresponden con las líneas de entrada de reloj y de datos respectivamente.

Patas 15 y 16 (OSC1/CLKIN y OSC2/CLKOUT): Corresponden a los *pines* de la entrada externa de reloj y salida de oscilador al cristal respectivamente. A través de ellos regularemos la velocidad a la que queremos que trabaje el PIC.

1.2. Arquitectura interna.

Actualmente, hay dos estructuras internas básicas que acostumbran a seguir la mayoría de sistemas integrados de control, que son la arquitectura Von Neumann y la Harvard.

La **arquitectura Von Neumann** dispone de una sola memoria principal donde son almacenados tanto los datos como las instrucciones. A esta memoria se accede mediante un sistema de buses único, es decir, por un mismo camino se envían las direcciones y se extraen los datos e instrucciones.

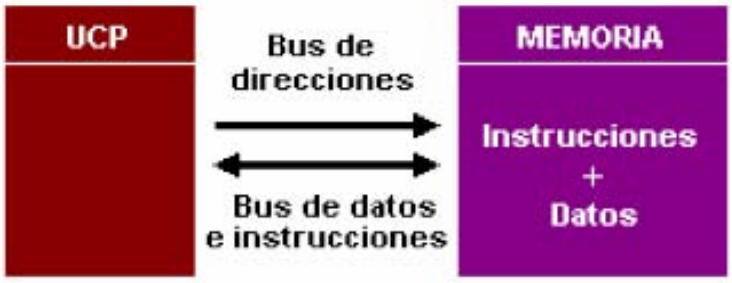


Figura 2: Esquema de la arquitectura Von Neumann.

La **arquitectura Harvard** en cambio, dispone de dos memorias independientes, una para los datos y la otra para las instrucciones. Cada una de estas memorias dispone de su propio sistema de buses de acceso, y es posible realizar operaciones de lectura o escritura simultáneamente en las dos memorias. Esta es la estructura utilizada para los microcontroladores PIC.

Figura 3: Esquema de la arquitectura Harvard.

1.3. Tipo de memorias.

Como hemos explicado en el apartado anterior, los microcontroladores PIC disponen de dos tipos de memorias: la de instrucciones y la de datos.

La **memoria de instrucciones**, también denominada memoria de programa, es la encargada de almacenar el código o programa que será ejecutado por el microcontrolador. Esta memoria está limitada por las características del PIC y no se puede ampliar de manera externa. Hay cinco tipos de memorias de instrucciones, de las cuales destacan dos:

- **Memoria EEPROM** (Electrical Erasable Programable Read Only Memory): Esta memoria es propia de PICs como el 16C84. La acción de acceder a ésta es realizada por un ordenador personal, mediante un circuito grabador. El número de veces que se puede grabar este tipo de memoria es finito, y está alrededor de las mil veces. Este tipo de memoria es relativamente lenta.

- **Memoria FLASH**: Cada vez es más utilizada y está reemplazando la EEPROM. Este tipo de memoria es la utilizada por el microcontrolador PIC 16F84A. Tiene las mismas características que la EEPROM pero consume una menor cantidad de energía y dispone de una capacidad de almacenado mayor.

La **memoria de datos** es la encargada de almacenar las variables de estado del programa y los procesos que se están llevando a cabo durante la ejecución del programa. Es un tipo de memoria temporal, en la cual la información se pierde al desconectar el sistema. Esta memoria se divide en dos: la memoria **RAM estática o SRAM**, donde se encuentran los registros, y la memoria **EEPROM**, dónde opcionalmente se pueden almacenar datos que no se pierdan al desconectar la alimentación.

1.4. Oscilador externo.

El oscilador externo, denominado también reloj, es un circuito que se encarga de determinar la velocidad a la que debe trabajar el PIC. En el PIC 16F84A se conecta a las patas 15 y 16. Es un circuito muy simple, pero imprescindible para el correcto funcionamiento del microcontrolador.

Hay cuatro tipos de osciladores aplicables al PIC 16F84A:

- **RC**: Oscilador compuesto por una resistencia y un condensador.
- **XT**: Oscilador formado por un cristal.
- **HS**: Oscilador formado por un cristal de alta velocidad.
- **LP**: Oscilador formado por un cristal de baja frecuencia y bajo consumo de potencia.

Los más utilizados son el RC y el XT. En el momento de la grabación del microcontrolador se tiene que especificar la clase de oscilador que utilizará.

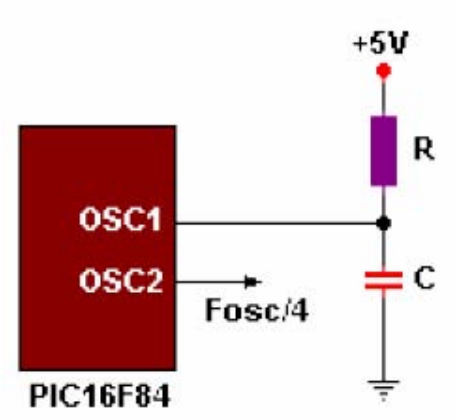


Figura 4: Esquema de un oscilador RC.

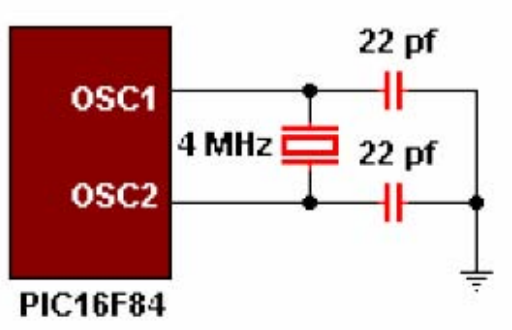


Figura 5: Esquema de un oscilador XT.

2. Programación de los microcontroladores PIC.

La programación del microcontrolador consiste en grabar en su memoria de programa el código que queremos que ejecute. Para realizar esta acción es necesaria la utilización de un hardware o elemento físico que nos permita conectar el microcontrolador a nuestro PC (programador), y de un software o programa informático que grabe nuestro código en la memoria del PIC.

Como programador utilizaremos el pipo2. Este programador ha sido diseñado por José Manuel García, y está basado en el programador "Ludipipo". Los esquemas eléctricos se pueden descargar gratuitamente de internet, al igual que el listado de componentes necesarios para su fabricación.
Su montaje es muy sencillo y debería funcionar correctamente en cualquier ordenador personal.

Para enviar nuestro código al programador usaremos el programa informático IC-Prog.
Este programa es uno de los más utilizados por los programadores de PICs, puesto que es una plataforma muy sencilla con la que trabajar, su comercialización es libre (lo que quiere decir que lo puedes conseguir gratuitamente en cualquier página web de internet) y se puede configurar para trabajar en castellano.
Este programa nos permite tanto grabar, como leer y borrar infinidad de microcontroladores PIC.
La manera de trabajar con el IC-Prog es muy sencilla. Lo primero que debemos hacer es seleccionar el tipo de programador que utilizaremos y el puerto del ordenador al que estará conectado. Después, el modelo de microcontrolador PIC. Una vez configuradas estas dos opciones, sólo queda seleccionar el tipo de oscilador con el que trabajará el PIC.

Tras realizar la configuración del programa para que funcione correctamente con la selección de elementos que utilizaremos, sólo nos falta abrir el archivo que queramos enviar a nuestro microcontrolador y darle al botón *grabar*.

2.1. El lenguaje de programación MikroBasic.

Actualmente existe un gran número de lenguajes de programación con los que poder escribir el código de nuestro robot, pero la mayoría son complejos y requieren de una buena base de programación.

El entorno MikroBasic nos permite programar microcontroladores de la marca Microchip utilizando el lenguaje *BASIC*, en un entorno muy visual y fácil de entender, con una gran cantidad de librerías que nos permitirán controlar con nuestro PIC infinidad de periféricos cómo pueden ser, por ejemplo, pantallas LCD.
La estructura básica de un programa se podría dividir en dos: las declaraciones, y el programa principal.

En las declaraciones se tienen que especificar todas las constantes y variables que se utilizarán, y de qué tipo serán, los procedimientos y las funciones.
En el programa principal se escribe el código que queremos que ejecute nuestro microcontrolador, la secuencia de órdenes que seguirá.

Como cualquier variable del lenguaje de programación *BASIC*, hay una serie de estructuras de control que nos facilitan la programación. Destacan la instrucción condicional “if”, que nos permite plantear condiciones como por ejemplo: si el sensor está activado, activa los motores, sino, que sigan parados; la instrucción “select case”, que nos permite formular un seguido de condiciones; la instrucción iterativa “for”, que nos permite crear bucles, como por ejemplo: desde que el valor de la variable sea 0 hasta que valga 100, los motores activados, cuando pase de 100, que se paren; y la instrucción “while” que nos permite formular bucles mientras una condición sea verdadera.

Para explicar detenidamente como se realiza un programa, haría falta profundizar más en el mundo de la programación, de las secuencias y procedimientos, subrutinas y operaciones que lleva a cabo un programa, pero en nuestro caso bastará con las mínimas nociones, puesto que será todo secuencial (una orden tras otra).

3. Desarrollo del proyecto.

3.1. Objetivos que debe alcanzar.

El objetivo de este proyecto es diseñar un autómata programable controlado por un microcontrolador PIC 16F84A, que sea capaz de trabajar de forma autónoma y realizar con éxito las tareas para las que será diseñado.

La función que deberá realizar será la de desplazar por la zona de trabajo dos cargas situadas sobre sus respectivos palés.

La carretilla iniciará el programa estando situada delante del primer palé. Ésta se desplazará hasta penetrar la carga por completo.

A continuación alzará las palas para levantar el palé de la superficie de la zona de trabajo y, después de una breve parada, retrocederá.

Una vez en el centro de la zona de trabajo, hará un giro de 90º a la derecha y se desplazará hasta el margen de la zona de trabajo dónde bajará las palas para dejar la carga.

Para finalizar la primera parte del programa, retrocederá hasta el medio de la zona de trabajo dónde volverá a hacer un giro de 90º hacia la derecha para encarar la segunda carga.

Una vez en esta posición, el autómata volverá a ejecutar el mismo programa con el fin de desplazar el segundo palé hasta el otro lado de la zona de trabajo.

3.2. Descripción del autómata.

El autómata programable seguirá el modelo de una carretilla mecánica, como las que se pueden encontrar en cualquier industria, pero con la limitación de que sus palas no podrán describir movimientos verticales, sino que sólo podrán bascular.

Ésta estará constituida por una base de madera, que hará la función de chasis, donde se unirán el resto de componentes.

La tracción la proporcionarán cuatro motores de corriente continua con sus respectivas cajas reductoras para disminuir su velocidad, conectados los dos de atrás con los dos de delante de su mismo lado.
Los ejes de éstos estarán unidos a cuatro ruedas de goma de 45 mm de diámetro.

La dirección la controlarán los cuatro motores antes mencionados. Girando los motores de un lado en un sentido y los otros en sentido contrario conseguiremos que se produzcan giros muy precisos sin que la carretilla se mueva de su situación en el espacio.

Las palas estarán hechas con una fina lámina de fibra de vidrio y estarán unidas al chasis mediante un eje que permitirá que estas basculen hacia adelante o hacia atrás.

La profundidad de las palas estará determinada por un sensor CNY70 compuesto por un emisor de infrarrojos y un fototransistor que hará parar la carretilla cuando la carga esté próxima al sensor.

Los diferentes movimientos que describirá la carretilla por la zona de trabajo estarán determinados por periodos de tiempos previamente determinados.

3.3. Entorno de trabajo.

El entorno en el que trabajará el autómata está diseñado específicamente para esta carretilla en concreto y el programa que ejecutará.

La superficie es de una madera muy fina y pulcra, cuadrada, de 1210 mm de lado.

Cada una de las dos cargas estará situada en uno de los lados, de manera opuesta como se muestra en el diagrama de la figura 6.

La carretilla iniciará el programa desde el centro de la zona de trabajo, encarada a la primera carga como también se muestra en la figura 6.

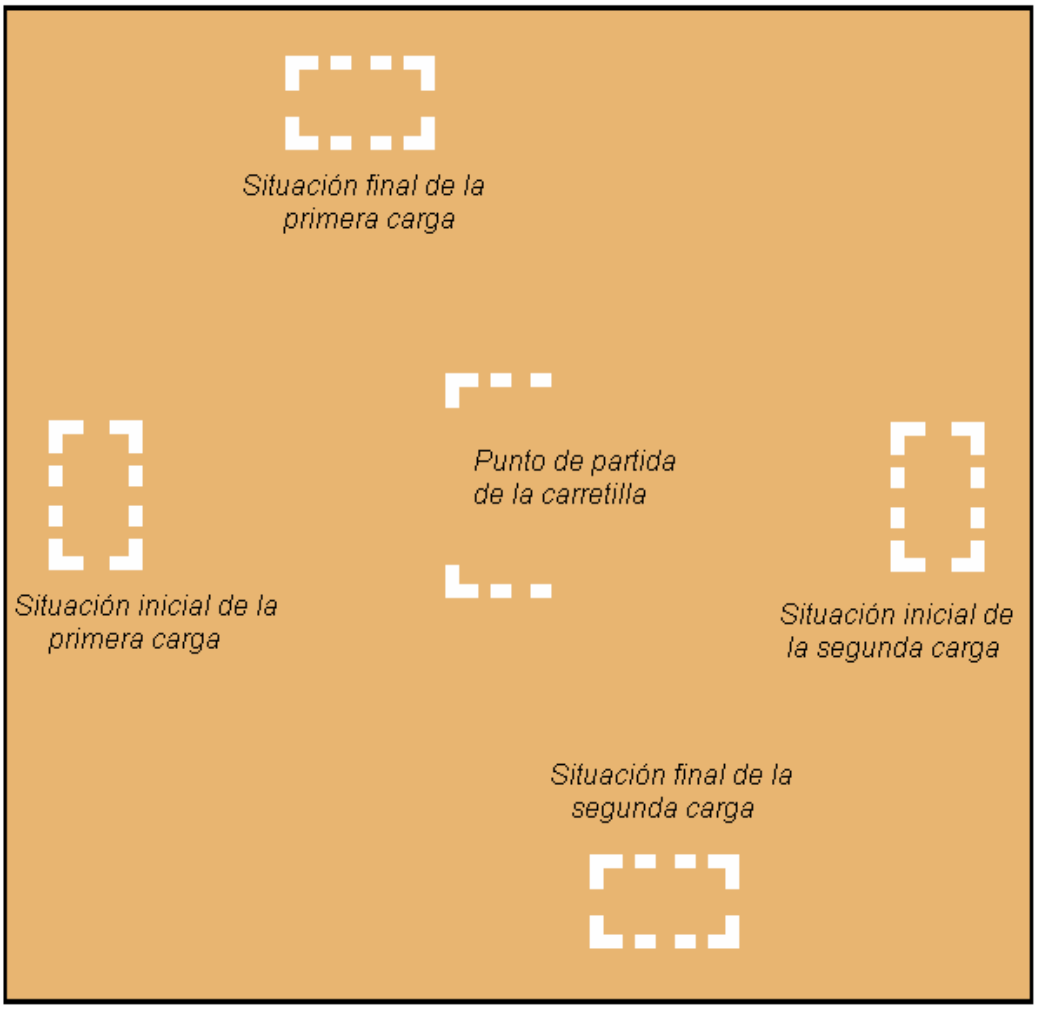


Figura 6: Diagrama de la zona de trabajo.

3.4. Diseño del circuito controlador.

El circuito controlador del autómata es un conjunto de circuitos combinados entre ellos, de forma que el microcontrolador puede, tanto enviar información a los actuadores, que en este caso serán los motores, como recibir información de su entorno mediante el sensor de infrarrojos.

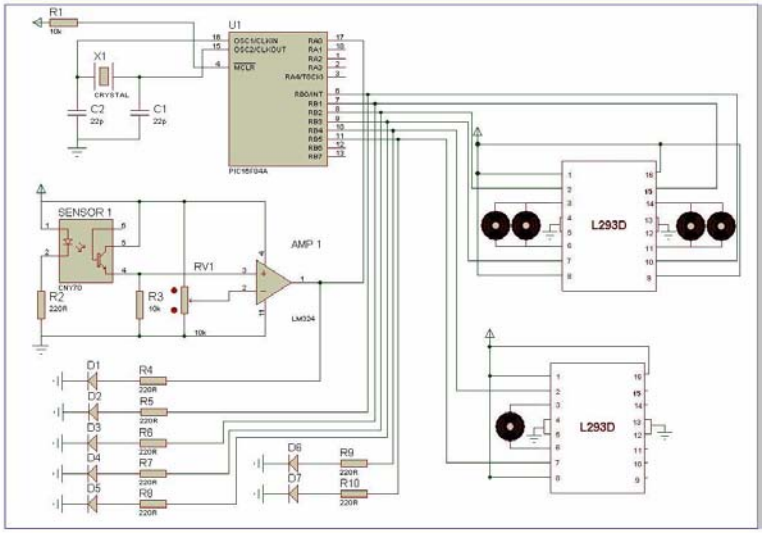


Figura 7: Esquema del circuito controlador.

3.4.1. Circuito de reset.

El PIC 16F84A tiene internamente un circuito temporizador conectado al *pin* 4 que se activa cuando llega una señal eléctrica a éste. El PIC se queda en estado de *reset* durante un breve periodo de tiempo, mientras se estabilizan todas las señales del circuito. Es imprescindible que esta pata esté conectada siempre a la misma tensión que el microcontrolador, ya que al conectar el circuito, el PIC empezará a ejecutar el programa guardado en su memoria desde el principio. También es recomendable colocar un pulsador en la entrada de este *pin* para poder hacer un *reset* en cualquier momento que fuera necesario, y que el programa empiece a ejecutarse desde el principio.

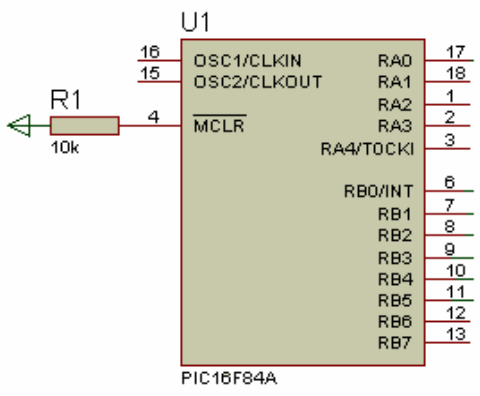


Figura 8: Diagrama del circuito de reset.

3.4.2. Circuito oscilador.

El circuito oscilador es el más importante en cualquier aplicación que se quiera realizar con un microcontrolador PIC. Éste se encarga de determinar la velocidad a la que trabajará el PIC, y sin él no funcionaría correctamente.

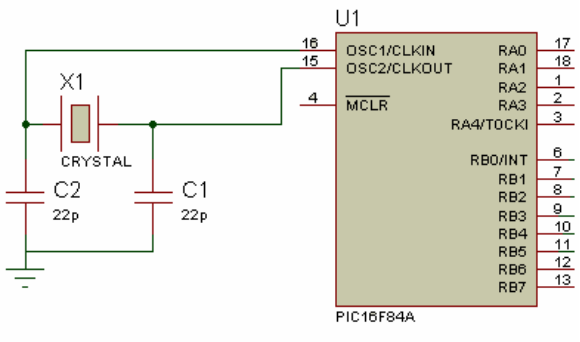


Figura 9: Diagrama del circuito oscilador.

Nosotros utilizaremos un oscilador XT formado por un cristal de cuarzo de 4 MHz y dos condensadores de 22pf. Este oscilador garantiza una mayor precisión y una buena arrancada del microcontrolador. Internamente esta frecuencia es dividida por cuatro, ejecutando de este modo una instrucción cada microsegundo.

3.4.3. Circuito detector de carga.

La función del circuito detector de carga es la de parar la carretilla cuando las palas de ésta penetren completamente el palé.

El elemento más importante de este circuito es el sensor CNY70. Este componente electrónico está compuesto por un emisor de infrarrojos y un fototransistor, que se encuentran uno junto al otro en la misma cápsula. El primero de estos elementos se encarga de emitir un rayo de luz que llega hasta el objeto que se quiere detectar. Este rayo llega reflejado al fototransistor, que al recibir la luz se satura y de esta manera deja pasar la corriente eléctrica.

La función del resto de elementos que forman el circuito detector es la de complementar el sensor CNY70 y hacer que este funcione correctamente. Las resistencias R2 y R3 de 220Ω y 10kΩ, protegen el emisor de infrarrojos y el fototransistor, respectivamente, de una tensión demasiado elevada.

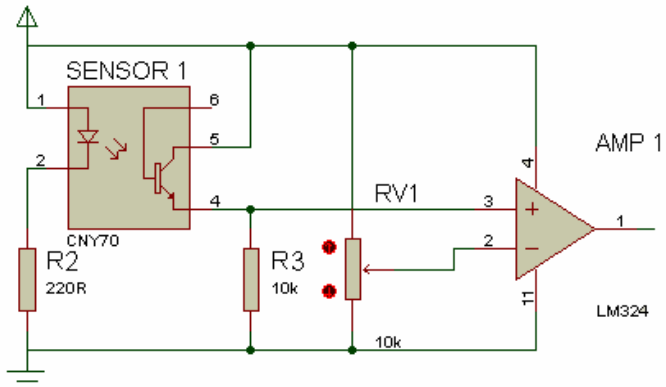


Figura 10: Diagrama del circuito detector de carga.

La resistencia variable RV1 determina el margen de activación del sensor.

Por último, el amplificador operacional AMP1, se encarga de enviar una señal útil de cinco voltios al microcontrolador, cuando el sensor esté activado. Este amplificador operacional se encuentra dentro de un circuito integrado denominado LM324, el cual contiene cuatro amplificadores operacionales.

3.4.4. Circuito controlador de motores.

Los elementos que controlarán los cinco motores de corriente continua que harán la función de actuadores, cuatro de ellos encargados de mover la carretilla por la zona de trabajo, y el otro de levantar las palas, serán dos circuitos integrados L293D.

Estos circuitos integrados son chips de dieciséis patas, que albergan en su interior cuatro canales, agrupados en dos grupos. Cada canal dispone de una entrada, de una salida y una línea de activación que activa los canales de dos en dos. Si quisiéramos activar y desactivar sólo los motores, sin tener en cuenta su sentido de giro, podríamos conectar hasta cuatro motores, cada uno de ellos en uno de los cuatro canales. Al pasar corriente eléctrica por la línea de activación y por la entrada del canal correspondiente, el motor conectado a la salida de este entraría en funcionamiento.

Ahora bien, nos interesa poder controlar también el sentido de giro, puesto que querremos que los motores que transmiten la tracción a las ruedas giren hacia un sentido o hacia el otro, y que el tercer motor, sea capaz de levantar o bajar la carga. Por esto conectaremos cada motor a las salidas de dos de los cuatro canales que tiene cada integrado L293D.
De esta manera, al pasar corriente eléctrica por la línea de activación se habilitarán los dos canales que controlan el motor. Si hacemos pasar corriente por una de las entradas, el motor girará en un sentido, y si hacemos pasar corriente por el otro, el motor girará en sentido contrario. Si las dos entradas tienen el mismo valor (las dos activadas o sin activar) el motor permanecerá inmóvil.

La disposición de los canales respecto a las dieciséis patas que tiene el circuito integrado es la siguiente:

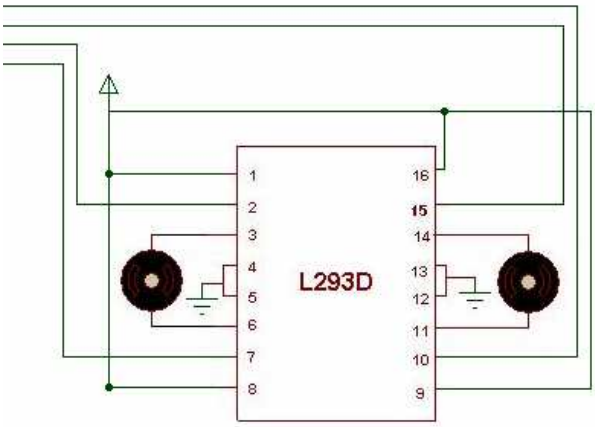


Figura 11: Diagrama del circuito controlador de motores.

• Las patas 1 y 9 son las líneas de activación de los canales 1, 2 y 3, 4 respectivamente.

• El *pin* 16 es la entrada de la alimentación del integrado.

• El *pin* 8 se encarga de suministrar la energía eléctrica a los motores.

• Las patas 4, 5, 12, 13 son las salidas a tierra.

• Por último, las patas 2, 7, 10 y 15 son respectivamente las entradas de los canales 1, 2, 3 y 4, y las patas 3, 6, 11 y 14 las salidas de los mismos canales.

3.4.5. Circuito diodos leds.

Esta parte del circuito no es necesaria para el correcto funcionamiento del autómata, pero si que nos será muy útil a la hora de identificar posibles errores.

Cada diodo led se conecta siempre en serie con una resistencia de 220Ω para protegerlo de una tensión demasiado elevada.
El primer diodo led está conectado a la salida del amplificador operacional situado en el circuito detector de carga. De esta manera, cada vez que se active el sensor, es decir, cada vez que detecte la carga, el diodo led se iluminará.

El resto de diodos leds están conectados a las salidas del microcontrolador que controlan los motores, así cada vez que el PIC emita una señal hacia los motores, el diodo led correspondiente nos informará de que el motor se tiene que poner en marcha.

Este circuito nos permitirá saber en todo momento si el programa se está ejecutando correctamente y en caso de que sucediera algún tipo de error, como que uno de los motores no girase, sabríamos si es debido a algún problema interno o de conexión del motor, o si el programa o el microcontrolador han fallado.

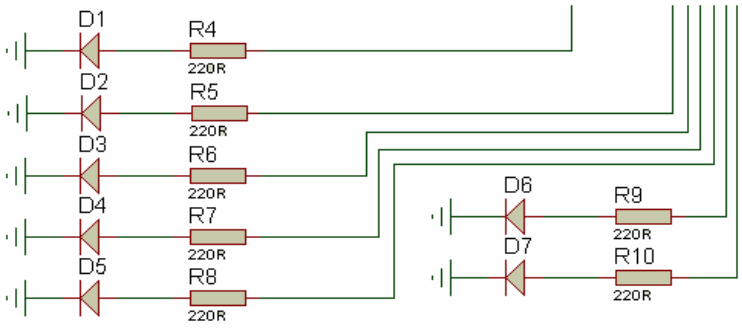


Figura 12: Diagrama del circuito diodos leds.

3.5. Diseño del programa.

El diseño de este programa está basado en los objetivos previos que se habían marcado para este proyecto. Como nuestra base de programación es casi nula, todo el programa ha sido diseñado de manera secuencial y sin atender ni tener en cuenta ningún tipo de interrupción externa al autómata.

A continuación está el programa que seguirá la carretilla para desplazar los dos palés por la zona de trabajo.

```
program Programa_esdelibro

TRISA = %11111

TRISB = %00000000

main:

while true

If PORTA=%00010 then

PORTB=%00000000
DELAY_MS(2000)
PORTB=%00010000
DELAY_MS(2000)
PORTB=%00000000
DELAY_MS(2000)
PORTB=%00001010
DELAY_MS(700)
PORTB=%00000000
DELAY_MS(2000)
PORTB=%00001001
DELAY_MS(2000)
PORTB=%00000000
DELAY_MS(2000)
PORTB=%00000101
DELAY_MS(700)
PORTB=%00000000
DELAY_MS(2000)
PORTB=%00100000
DELAY_MS(2000)
PORTB=%00000000
DELAY_MS(2000)
PORTB=%00001010
DELAY_MS(700)
PORTB=%00000000
DELAY_MS(2000)
PORTB=%00001001
DELAY_MS(2000)
PORTB=%00000000
DELAY_MS(2000)

else

PORTB=%00000101

end if

wend

end.
```

Figura 13: Imagen del programa diseñado.

Obviamente este programa requiere de las explicaciones pertinentes para poder llegar a comprenderlo. Lo analizaremos por partes, línea a línea, para poder entender que hará el autómata en cada momento del proceso.

De la línea 1 a la 11:

En la primera línea de programa se escribe la orden "program" seguida del nombre del programa.

En las líneas 3 y 5 configuramos las patas del PIC. A las del puerto A le damos valor 1 y a las del puerto B valor 0 para configurarlas como entradas y salidas respectivamente.

```
1  program Programa_esdelibro
2
3  TRISA = %11111
4
5  TRISB = %00000000
6
7  main:
8
9  while true
10
11 If PORTA=%00010 then
```

Figura 14: Imagen de la primera parte del programa.

Una vez configuradas las patas del microcontrolador, podemos pasar a escribir las instrucciones que deberá seguir.
Para introducir las instrucciones escribimos la orden "main:" y a continuación la secuencia de órdenes.

La primera estructura de control que escribimos es la de un "while". Con esta orden abrimos un bucle que hará que se repita el programa que hay escrito desde la línea 10 hasta la 49, siempre y cuando las declaraciones iniciales se cumplan (en este caso siempre se cumplirán, la configuración inicial de las patas es la misma para todo el programa).
A continuación escribimos otra estructura de control, la de "If". Esta orden expresa una condición, y expresado en términos coloquiales seria:

- Si, en este caso, la entrada 1 del puerto A tiene valor 1, entonces, realiza las instrucciones que hay a continuación, si no, haz lo que hay a continuación del "else".

En un principio, la entrada 1 del puerto A, que será el sensor del circuito detector, tiene valor 0, puesto que el sensor no está activado. Al no cumplirse la condición, el programa pasa a ejecutar lo que hay tras el "else", que es darle valor 1 a las salidas 0 y 2 del puerto B, dónde están conectados los polos positivos de los motores situados en las ruedas motrices.
La carretilla se desplazará en línea recta hacia delante hasta penetrar la carga y activar el sensor, momento en el que el programa pasará a ejecutar las instrucciones que hay a continuación del "then".

De la línea 13 a la 42:

En la línea trece es dónde realmente empiezan las instrucciones que seguirá la carretilla.

Todas las instrucciones están escritas de manera secuencial, una detrás de otra, separadas por un determinado intervalo de tiempo.

• En la línea 13, después de haber penetrado la carga con las palas, le damos valor 0 a todas las salidas del puerto B, para mantener la carretilla en reposo.

• En la línea 14 determinamos el tiempo de esta parada, que será de 2000 milisegundos.

• En las líneas 15 y 16 le damos valor 1 a la quinta pata del puerto B, dónde está conectado el polo positivo del motor de las palas, para hacer que estas suban la carga durante dos segundos.

• Después de una parada de dos segundos, hacemos retroceder la carreta durante 700 milisegundos dándole valor 1 a las patas 2 y 4 dónde están conectados los polos negativos de los motores de las ruedas motrices.

```
 ▪
 ▪ PORTB=%00000000
 ▪ DELAY_MS(2000)
15 PORTB=%00010000
 ▪ DELAY_MS(2000)
 ▪ PORTB=%00000000
 ▪ DELAY_MS(2000)
 ▪ PORTB=%00001010
20 DELAY_MS(700)
 ▪ PORTB=%00000000
 ▪ DELAY_MS(2000)
 ▪ PORTB=%00001001
 ▪ DELAY_MS(2000)
25 PORTB=%00000000
 ▪ DELAY_MS(2000)
 ▪ PORTB=%00000101
 ▪ DELAY_MS(700)
 ▪ PORTB=%00000000
30 DELAY_MS(2000)
 ▪ PORTB=%00100000
 ▪ DELAY_MS(2000)
 ▪ PORTB=%00000000
 ▪ DELAY_MS(2000)
35 PORTB=%00001010
 ▪ DELAY_MS(700)
 ▪ PORTB=%00000000
 ▪ DELAY_MS(2000)
 ▪ PORTB=%00001001
40 DELAY_MS(2000)
 ▪ PORTB=%00000000
 ▪ DELAY_MS(2000)
```

Figura 15: Imagen de la segunda parte del programa.

• Una vez hecha la siguiente parada de dos segundos, damos valor 1 a la primera y cuarta pata del puerto B, dónde están conectados los polos positivo y negativo de los motores de la izquierda y de la derecha respectivamente, para describir un giro hacia la derecha de 90º.

• A continuación, en las líneas 27 y 28, hacemos mover la carretilla durante 700 milisegundos para situarla en la posición dónde queremos que deje la carga.

• En las líneas 31 y 32 le damos valor 1 a la sexta pata del puerto B durante dos segundos para hacer que las palas bajen y dejen la carga sobre la zona de trabajo.

• Después de otra parada de dos segundos determinada en las líneas 33 y 34, hacemos retroceder la carretilla durante 700 milisegundos dándole valor 1 a las patas 2 y 4 del puerto B.

• Para finalizar la secuencia y dejar la carretilla encarada a la segunda carga, hacemos girar el autómata 90º a la derecha dándole valor 1 a las patas 1 y 4 del puerto B durante dos segundos.

De la línea 13 a la 42:

• En la línea 44 escribimos la orden “else” que hará que el PIC ejecute lo que hay escrito en la línea 46 cuando el sensor esté desactivado.

• En la línea 46 le damos valor 1 a las patas 1 y 3 del puerto B dónde están conectados los polos positivos de los motores que proporcionan la tracción a las ruedas, para hacer que la carreta se desplace hasta encontrar la carga y activar el sensor.

• En la línea 48 cerramos la instrucción de control “If”.

• En la línea 50 cerramos el bucle iniciado con “el while”.

• En la línea 52, para finalizar el programa, escribimos la instrucción “end”.

```
else
45
PORTB=%00000101
48 end if
50 wend
end.
```

Figura 16: Imagen de la tercera parte del programa.

3.6. Parte mecánica.

La parte mecánica de la carretilla ha sido diseñada para cumplir las funciones y objetivos que previamente se habían marcado.

Una base de madera de 5 mm de grosor será el chasis, en el que se unirán el resto de componentes. La elección de este material para la construcción del chasis ha sido debido a su rigidez, necesaria para soportar el peso de los motores, las baterías y las palas, y por la facilidad con la que se trabaja este material, que nos facilitará la acción de cortarlo y perforarlo.

Las palas serán el elemento más importante de la carretilla, encargadas de levantar y transportar los palés. Éstas estarán construidas a partir de una fina lámina de fibra de vidrio, unidas a un eje metálico que permitirá su libre movimiento.

El sensor CNY70 del circuito detector de carga estará situado sobre las palas, en el centro, unido a estas mediante una pieza mecánica.

Los motores encargados de proporcionar la tracción a la carretilla son motores de corriente continua de 4,5 V, acompañados de sus respectivas cajas reductoras para reducir su velocidad y que la carretilla se mueva mediante movimientos controlados y precisos. Las ruedas motrices situadas en los ejes de los motores son ruedas de goma de 45 mm de diámetro, extraídas del juego de construcción “MECCANO”.

3.7 Listado de componentes.

A continuación está el listado de todos los componentes necesarios para la realización de este proyecto:

Parte mecánica:

- Plancha de madera de 210 x 140 mm y 5 mm de grosor.
- 4 motores de corriente continua de 4,5 V con caja reductora.
- 1 motor de corriente continua de 4,5 V con caja más reductora.
- 4 ruedas de goma de 45 mm de diámetro.
- 1 eje de 60 mm de longitud y 4 mm de diámetro.
- 4 varillas roscadas de 70 mm de longitud y 4 mm de diámetro.
- Plancha de fibra de vidrio de 200 x 150 mm y 2 mm de grosor.
- Varias piezas mecánicas del juego de construcción “MECCANO”.
- 1 cadena de 30 mm de longitud.
- 2 ruedas dentadas para estas cadenas.
- 1 porta pilas de tres pilas AAA.
- Varios tornillos y tuercas del juego de construcción “MECCANO”.

Circuito controlador:

- Placa perforada para circuitos electrónicos.
- 3 pilas AAA.
- 1 microcontrolador PIC 16F84A.
- 2 circuitos integrados L293D.
- 1 circuito integrado LM324.
- 1 cristal de cuarzo de 4MHz.
- 2 condensadores de 22pF.
- 8 resistencias de 220Ω.
- 2 resistencias de 10kΩ.
- 1 resistencia variable de 10kΩ.
- 7 diodos leds.
- 7 regletas para circuito impreso.
- Cable para conexiones.
- 1 conmutador.